



13 ファイルI/O



Created by GT F

Last updated 2019-08-18

- [File Object](#)
 - [ファイルのパス](#)
 - [ディレクトリ](#)
 - [ファイル](#)
- [代表な操作](#)
- [特定タイプのファイルを取得する](#)
- [Stream](#)
- [Text File](#)
 - [文字コード](#)
 - [読み取り](#)
 - [書き込み](#)
- [ファイル操作](#)
 - [固定長ファイル](#)
 - [区切ファイル](#)
 - [プロパティファイル](#)
- [try resources](#)
- [質問](#)
- [補足](#)
 - [代表的なIOクラス体系](#)

File Object

ファイルのパス

Linux、Windows、Mac X

ディレクトリ

ディレクトリ（Directory）は特殊のファイルです。ディレクトリの別名はフォルダー（Folder）と呼びます。

ファイル

代表な操作

ファイルに代表する操作はすべて覚えてください。ディレクトリは特殊のファイルです。

メソッド	説明
boolean exists()	ファイル存在有無チェック
boolean isDirectory()	ディレクトリを判断する
boolean createNewFile()	ファイルを作成
boolean mkdir()	ディレクトリを作成する
boolean mkdirs()	複数のディレクトリを作成する
boolean delete()	ファイルを削除
File[] listFiles()	ディレクトリの下位ファイルを取得する
String getAbsolutePath()	ファイルの絶対パスを取得する
String getName()	ファイル名を取得する

特定タイプのファイルを取得する

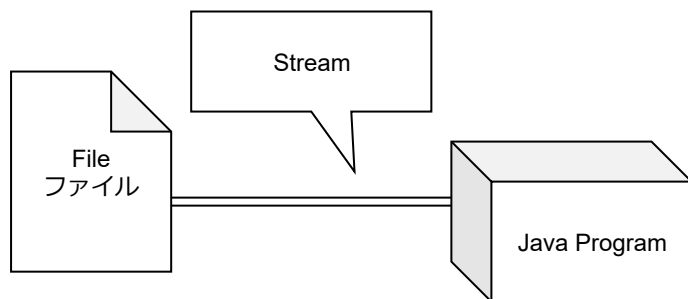
メソッド `listFiles(FileFilter ff)` の引数FileFillterを渡して、特定なファイルを取得可能です。

以下例で、homeディレクトリの下にExcelファイルのみを取得する。

```
1 public class Kicker {
2
3     public static void main(String... args) {
4         File[] excels = new File("home").listFiles(new FileFilter() {
5             @Override
6             public boolean accept(File file) {
7                 return file.getName().endsWith(".xlsx") // Excel 2007 以後
8                     || file.getName().endsWith(".xls"); // Excel 97 以後
9             }
10        });
11    }
12 }
```

Stream

Javaはファイルにデータを書き込み又はファイルからデータの読み取りはすべてストリーム（Stream）に介してます。ストリームとは、データの流れ道です。



ストリームの操作は以下 3 ステップがあります。

1. ストリームを開く
2. 操作
3. ストリームを閉じる

例：ファイルのストリームを開いて、20バイトを読み取った後、閉じます。

```
1  import java.io.File;
2  import java.io.FileInputStream;
3  import java.util.logging.Level;
4  import java.util.logging.Logger;
5
6  public class FileStreamSample {
7
8      public static void main(String...args) {
9          try {
10             File file = new File("C:\\Test\\data.dat");
11             FileInputStream in = new FileInputStream(file); // 1. ストリームを開く
12             byte[] data = new byte[20];
13             in.read(data); // 2. データ操作
14             in.close(); // 3. ストリームを閉じる
15         } catch (Exception ex) {
16             Logger.getLogger(FileStreamSample.class.getName()).log(Level.SEVERE, null, ex);
17         }
18     }
19
20 }
```

※すべてのIO操作例外発生する可能性があります。try catch 文で例外を処理する必要があります。

Text File

文字コード

読み取り

BufferedReaderを使用して、テキストファイル行数分を読み取ります。

```
1  try {
2      BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream("C:\\data.txt"),
3          "UTF-8"))
4      String line;
5      int i = 0;
6      while ((line = br.readLine()) != null) {
7          System.out.println(line);
8          i++;
9      }
10     br.close();
11 } catch (IOException e) {
12     e.printStackTrace();
13 }
```

書き込み

BufferedWriterを使用して、テキストファイルを書き込みます。

```
1  try {
2      Writer out = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("C:\\data.txt"),
3          "UTF-8"));
4      out.write("DCNet Java 教育");
5      out.close();
6 } catch (IOException e) {
7     e.printStackTrace();
8 }
```

ファイル操作

固定長ファイル

区切ファイル

CSV（comma-separated values）いくつかのフィールド（項目）を区切り文字であるカンマ「,」で区切ったテキストデータおよびテキストファイル。拡張子は .csv、MIMEタイプは text/csv。

区切り文字	
改行コード	
囲文字	

プロパティファイル

プロパティファイルとは：

読み取り

```
1 try (InputStream input = new FileInputStream("path/to/config.properties")) {
2     Properties prop = new Properties();
3     prop.load(input);
4     System.out.println(prop.getProperty("db.url"));
5     System.out.println(prop.getProperty("db.user"));
6     System.out.println(prop.getProperty("db.password"));
7 } catch (IOException ex) {
8     ex.printStackTrace();
9 }
```

書き込む

```
1 try (OutputStream output = new FileOutputStream("conf/db.properties")) {
2     Properties prop = new Properties();
3     prop.setProperty("db.url", "localhost");
4     prop.setProperty("db.user", "postgres");
5     prop.setProperty("db.password", "postgres");
6     prop.store(output, null);
7 } catch (IOException io) {
8     io.printStackTrace();
9 }
```

try resources

質問

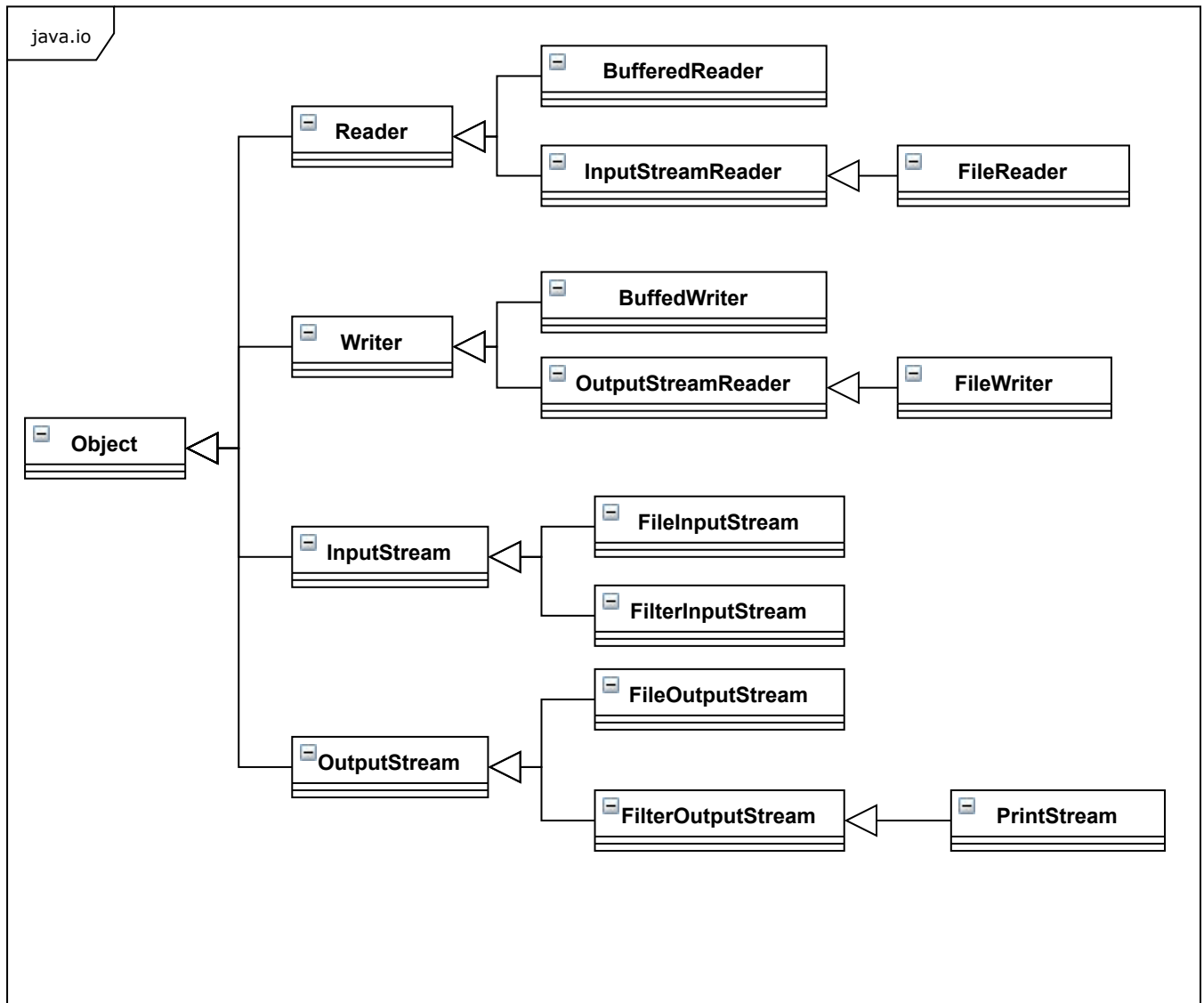
質問 1 : ログ分析アプリ作成してください。

質問 2 : CSV編集ツールを作成してください。

質問 3 :

補足

代表的なIOクラス体系



 Like Be the first to like this

No labels 